TECHNICAL REPORT
NATICK/TR-12/019

AD ________________

ALGORITHMS FOR GROUND SOLDIER BASED SIMULATIONS AND DECISION SUPPORT APPLICATIONS PHASE I

by
Steve Borkman
Marlo Verdesca
and
Jon Watkins

Dignitas Technologies, LLC
Orlando, FL 32817

May 2012

Final Report
January 2011 – July 2011

Approved for public release; distribution is unlimited

Prepared for
U.S. Army Natick Soldier Research, Development and Engineering Center
Natick, Massachusetts 01760-5020

UNCLASSIFIED

DISCLAIMERS

The findings contained in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of trade names in this report does not constitute an official endorsement or approval of the use of such items.

DESTRUCTION NOTICE

For Classified Documents:

Follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For Unclassified/Limited Distribution Documents:

Destroy by any method that prevents disclosure of contents or reconstruction of the document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 07-05-2012	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* January 2011 –July 2011

4. TITLE AND SUBTITLE

ALGORITHMS FOR GROUND SOLDIER BASED SIMULATIONS AND DECISION SUPPORT APPLICATIONS--PHASE I

5a. CONTRACT NUMBER
W911QY-11-P-0080

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Steve Borkman, Marlo Verdesca, and Jon Watkins

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Dignitas Technologies, LLC
3504 Lake Lynda Drive, Suite 170
Orlando, FL 32817

8. PERFORMING ORGANIZATION REPORT NUMBER
1101208-003R2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Natick Soldier Research, Development and Engineering Center
ATTN: RDNS-TSM (T. Gilroy)
Kansas Street, Natick, MA 01760-5020

10. SPONSOR/MONITOR'S ACRONYM(S)
NSRDEC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NATICK/TR-12/019

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Report developed under Small Business Innovation Research contract. This report describes the results of the research conducted by Dignitas Technologies, LLC, from January 2011 to July 2011, under the Algorithms for Ground Soldier Based Simulations and Decision Support Applications (Tactical Aids) Phase I Small SBIR contract awarded by the Natick Soldier Research, Development and Engineering Center. Dignitas identified, designed and documented detailed methodology descriptions for two terrain reasoning algorithms (Agent Route Planning and Agent Overwatch Position) meant to enhance both constructive simulation (situations traditionally heavily scripted in extensive detail) and operational capabilities (capabilities delivered to aid our Warfighters). The team created many supporting analysis artifacts and also implemented a prototype graphical user interface (GUI) to help illustrate how the complex data input requirements for these services would be met. A secondary artifact, albeit less tangible, was an enhanced understanding of IWARS functionality and capabilities, which provided insight into how other Army applications can benefit IWARS and vice versa.

15. SUBJECT TERMS

ROUTING VERIFICATION OVERWATCH POSITION SCU(SMALL COMBAT UNITS)
INFANTRY METHODOLOGY TERRAIN INTELLIGENCE DECISION SUPPORT SYSTEMS
VALIDATION ROUTE PLANNING TACTICAL DECISION AIDS GRAPHICAL USER INTERFACE
ALGORITHMS TERRAIN SERVICES SOLDIER-CENTRIC DESIGN COMPUTERIZED SIMULATION
SBIR REPORTS MILITARY PLANNING IWARS(INFANTRY WARRIOR SIMULATION)

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Thomas Gilroy
U	U	U	SAR	40	**19b. TELEPHONE NUMBER** *(include area code)* (508) 233-5855

This page intentionally left blank

Table of Contents

Table of Figures

This page intentionally left blank

ALGORITHMS FOR GROUND SOLDIER BASED SIMULATIONS AND DECISION SUPPORT APPLICATIONS

1 Introduction

This report describes the results of the Algorithms for Ground Soldier Based Simulations and Decision Support Applications (Tactical Aids) Phase I Small Business Innovation Research (SBIR) effort conducted by Dignitas Technologies, LLC from January 2011 to July 2011. The Tactical Aids Phase I SBIR is a Natick Soldier Research, Development and Engineering Center (NSRDEC) research effort intended to develop and demonstrate algorithms and methodologies that enhance analysis driven, ground soldier-centric constructive simulations and provide a proof of concept for small combat unit decision support applications.

The primary artifacts produced from this effort include detailed methodology descriptions for two terrain reasoning algorithms (Routing and Overwatch), a prototype graphical user interface (GUI) implementation to help illustrate how the complex data input requirements for these services would be met, and supporting analysis artifacts. These technical artifacts are expanded upon in Section 2. A secondary artifact, albeit less tangible, was an enhanced understanding of Infantry Warrior Simulation (IWARS) functionality and capabilities, which gives greater insight into how other Army applications can benefit IWARS and vice versa. From this insight, possible next steps for this work are discussed in Section 4.3, and NSRDEC's emphasis on Verification and Validation (V&V), is described in Section 4.2.

This Phase I SBIR represented Dignitas' first opportunity to work directly with NSRDEC personnel. In light of this, it was inevitable that the submitted proposal did not fully align with NSRDEC's objectives, and thus objectives needed to be reprioritized. Basically, focus was shifted to the initial artifacts for methodology definition, and away from prototyping. The original overall plan can still be carried out through a Phase II effort. Some of the deviations from the original proposal are described in Section 4.1. Readers should reference the Phase I proposal for details on the overall vision.

While Dignitas personnel have extensive experience in Army Computer Generated Forces (CGFs), they previously had only limited experience with IWARS. An initial analysis was conducted to better understand IWARS capabilities. IWARS is somewhat different from large Army CGF systems, such as One Semi-Automated Forces (OneSAF), Close Combat Tactical Trainer (CCTT) Semi-Automated Forces (SAF), or OneSAF Testbed Baseline (OTB) and thus it was important to understand those differences in order to capture algorithm descriptions that fit in well with IWARS' current paradigms.

Throughout Phase I, draft material was presented for review and feedback. NSRDEC's feedback was very helpful, both on general structure and on content, and especially insights into what IWARS presently handles as well as military subject matter expertise. This feedback was incorporated into the final methodologies documents and prototype GUI implementation, and Dignitas would welcome the opportunity to implement these methodologies during Phase II.

2 Technical Approach

The core focus of the Phase I effort was to identify, design and document methodologies that will both enhance constructive simulation (situations traditionally heavily scripted in extensive detail) and operational capabilities (capabilities delivered directly to and aiding Warfighters). These capabilities, which will evolve from design to implementation in Phase II, will be leveraged to enhance the automation and execution of individual agents and/or small units across both targeted domains. The end goal of this project is a single methodology for multiple domains.

Considering that the team was new to IWARS, this project began with an IWARS research and fact finding effort. This effort was performed to better understand general modeling and simulation M&S needs, as well as overall technical capabilities and objectives of IWARS. The team accomplished this by reading through the available IWARS documentation, and by installing and running the IWARS system at the Dignitas facility. Roger Schleper, NSRDEC, also provided information regarding IWARS and how it handles terrain data and terrain related services.

The initial step of methodology development was identifying the targeted methodologies. Work focused on identifying services beneficial for single agents/small units in both constructive and operational domains, and that were non-existent in IWARs (the target constructive simulation of the customer). The main focus was on capabilities that usually require substantial human in the loop (HITL) interaction, but could be algorithmically determined based on the terrain model/services and other simulation factors. During this identification phase, conversations took place with the customer to ensure the services that they felt were most needed were considered. From this analysis and customer interaction two methodologies were targeted: agent overwatch position and agent route planning.

Prior to defining the methodologies, key components were identified to serve as the basis for algorithm design. A service is only as good as its underlying model and inputs, the classic “garbage in, garbage out” problem. Before tackling methodology design, both the use cases of the service (constructive and operational) and the input components required for the algorithm were identified. Detailed use case and input component information can be found within the following documents: Agent Route Planner (ARP) Methodology, Agent Overwatch Position (AOP) Methodology and Tactical Aids Algorithm Factors.

A major consideration during methodology design was what results were going to be provided to the user. The decision was, based upon interaction with the customer, to make a guiding principle to not just provide a “right answer”, but a set of scored, “good” results that the user could use to help them make their own decision as to which is the “right” solution. Each service result will be supplied to the user with applicable meta-data explaining the rationale of the score.

Almost as important as the results of a service is how the results will be presented to the user. A convoluted user experience will result in indecipherable results and/or frustrated users, which eventually result in a service that is not used, and, therefore, of no benefit. Design of the GUI was considered of utmost importance. Both methodologies took user interaction and display of service results into consideration as they were designed. In fact, an important deliverable of this effort is a mocked up user interface effort which is discussed in Section 3.2.

3 Results

3.1 Methodologies

Both the AOP and ARP methodologies are complicated services and rely on varying factors in order to meet the needs of any given user at any time. These factors could range from simulated factors (location of enemy, fidelity of the underlying model) to real factors (performance requirements, current set of user requirements). In short, it is nearly impossible to design a single solution that could meet the varying needs of the community (especially considering targeting both the constructive simulation and operational communities).

Because of the variant nature of the problem, focus was given not only to designing a solution for the selected methodologies, but also the creation of a composable architecture allowing the user the flexibility to select (based on their needs) the factors used to calculate the solution set. The architecture also allows for future factor extension development. At any point a user can develop a factor that meets the defined Application Programming Interface (API) to calculate the results for the methodology.

The methodology results are calculated by the factors (and their weights) selected and defined by the user. The user has the capability to configure their instance of the service prior to execution. They will be able to define, add, and select which factors will execute for their query. Beyond that, the user will be able to add a weight to each factor to add priority to the factors that are the most important. The user will also be able to define a minimum score to a factor, which will completely eliminate a query if the calculation falls below the minimum score. From a user interface standpoint, factors may be presented with checkboxes on the side, allowing an operator to add or remove factors as needed. The dialog could also provide a means for operators to assign a ranking between the factors based upon their objectives.

The core methodologies for both routing and overwatch position are essentially a different interface display, and factor suite, on the same methodology engine.

Prior to designing the architecture and default factors for the services, research was conducted internal and external to the simulation domain for current data structures, services, and algorithms which could be used in the design and development of the methodologies. Routing and overwatch position algorithms were found and examined in current constructive/virtual simulation systems including OneSAF, CCTT, and the Naval Post Graduate School. High-resolution visibility tests essential for both methodologies were studied, including: raster based line of sight, shadow mapping, shadow volumes, and depth map testing. Overwatch position calculation methods from different computer science domains (security camera placement algorithms, raster-based location selection, “the security guard problem” from academic computer science, and particle swarm optimization and other genetic algorithms) were investigated. In addition, industry standard routing solutions were assessed, including A* network traversal. Many artifacts from the analysis were uploaded to Basecamp for reference. Basecamp is a web-based project management and collaboration tool. Two methodology definitions were developed as part of this Phase I effort: ARP Methodology and AOP Methodology. They are included in this report as Appendix A and Appendix B, respectively.

3.2 Prototype demonstration

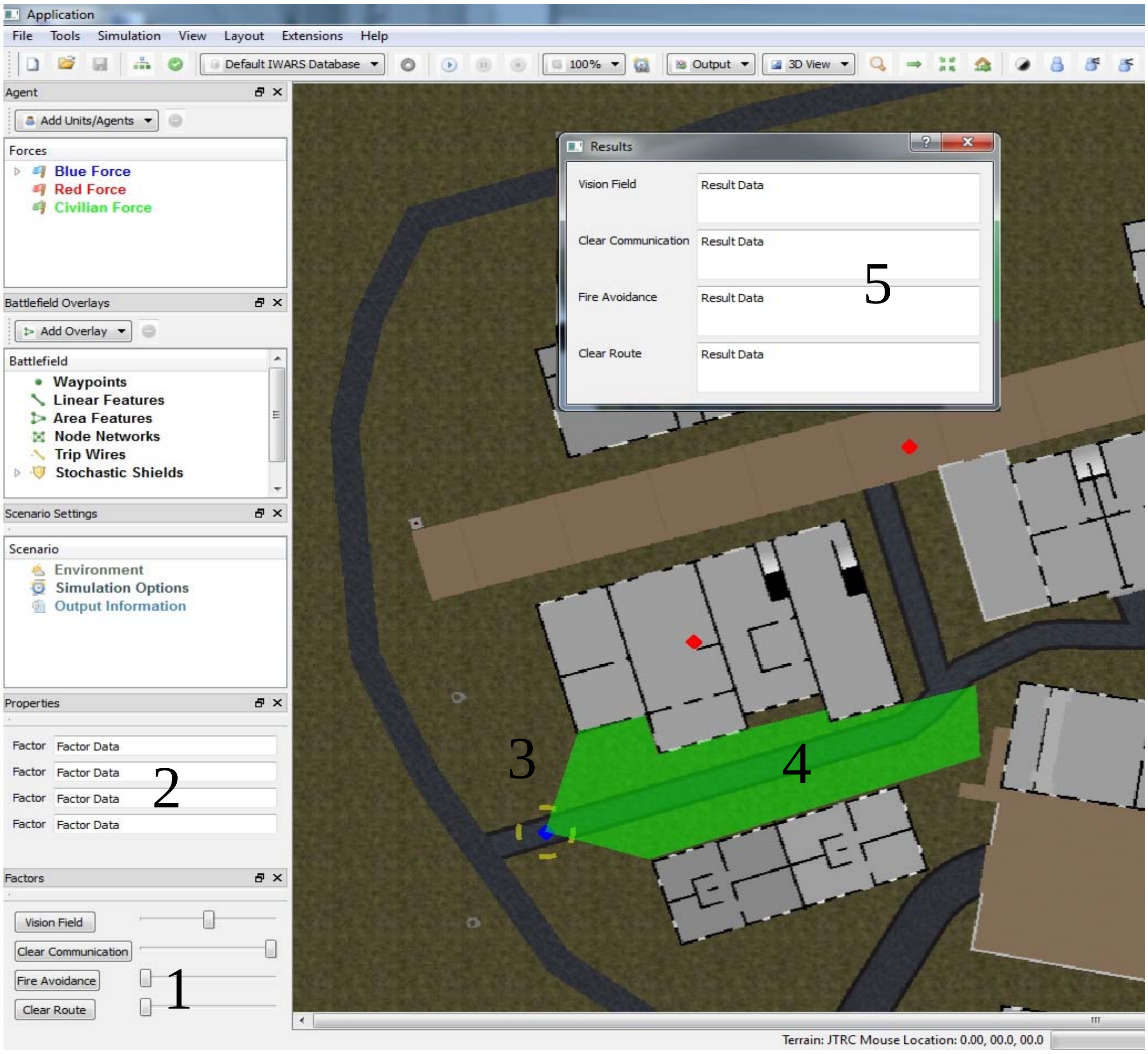
The following section describes the envisioned user interface for the newly defined methodologies: agent overwatch position and agent routing. Given the similar, composable architecture of the new services, there are significant similarities in the user interface for both services. For the purpose of this demonstration, focus was on the agent overwatch methodology. This feature will allow a user to predetermine an optimal set of locations for troop placement within a selected area of operation.

For demonstration purposes, a prototype was constructed for the proposed features. Figure 1 displays the reconstructed IWARS interface, with the agent overwatch service and routing service integrated into the GUI. The GUI displays a proof of concept model only and is not a fully functional set of services. A set of service algorithms and GUI design implementations will need to be made to properly represent the additional services.

This section explains in greater detail, Figure 1 and Figure 2 which clarifies how the system will work when fully implemented and operational. The agent overwatch positions will be determined by a set of inputs that determine the optimal location for entity placement. These inputs will be composed of factors (#1) such as vision field, clear communication, fire avoidance, and clear route. Each factor will have a set of sub properties (#2) that determine the overall make of a factor. The terrain display will predetermine a set of optimal locations for entity placement (#3) based on a selected area of operation. As factors are adjusted, the optimal entity location will be highlighted by an animated target system. The user can then right click on the optimal location to display a line of sight fan (#4), and resultant data (#5) for why that location was selected as the optimal entity placement location.

> (#1) Factors - Set of weights that determine the weighted factor of properties associated with the selected factor.
>
> (#2) Properties - Set of properties that are associated with a designated factor. Currently the user interface only displays generic properties. With a finished product in hand the property widget will display valid properties.
>
> (#3) Highlight Target System - Animated graphic easily displays the optimal entity placement location that the agent overwatch algorithm has selected based upon user property inputs and weighted factor selection.
>
> (#4) Line of Sight Fan – Shown after the user right clicks the optimal location. The system will display a line of sight fan that will visibly display the entity’s unobstructed line of sight view from the specified location to a specified radial distance. This distance will be defined by one of the properties.
>
> (#5) Resultant Data – Shown after the user right clicks the optimal location. The system will display a pop up dialog box that will display the resultant data for an optimal entity placement location. The resultant data will explain why that location was selected. The

dialog box will contain an explanation of each factor’s resultant data and how the factor affects the specified location.



Vision Field: Result Data

Clear Communication: Result Data

Fire Avoidance: Result Data

Clear Route: Result Data

Factor: Factor Data

Factor: Factor Data

Factor: Factor Data

Factor: Factor Data

Terrain: JTRC Mouse Location: 0.00, 00.0, 00.0

Figure 1: Overwatch Algorithm – IWARS

Figure 2 illustrates the route planner component for an agent’s position based on weighted factors. For the prototype, the model allows the user to display two different route plans based on the “Clear Route” factor for a single predetermined agent position. When the predetermined agent is selected as the optimal agent overwatch, the user can set the weight of the “Clear Route” factor to display a different route for each factor weight.

(#6) Route Plan – The system will display a possible route for the predetermined agent position.

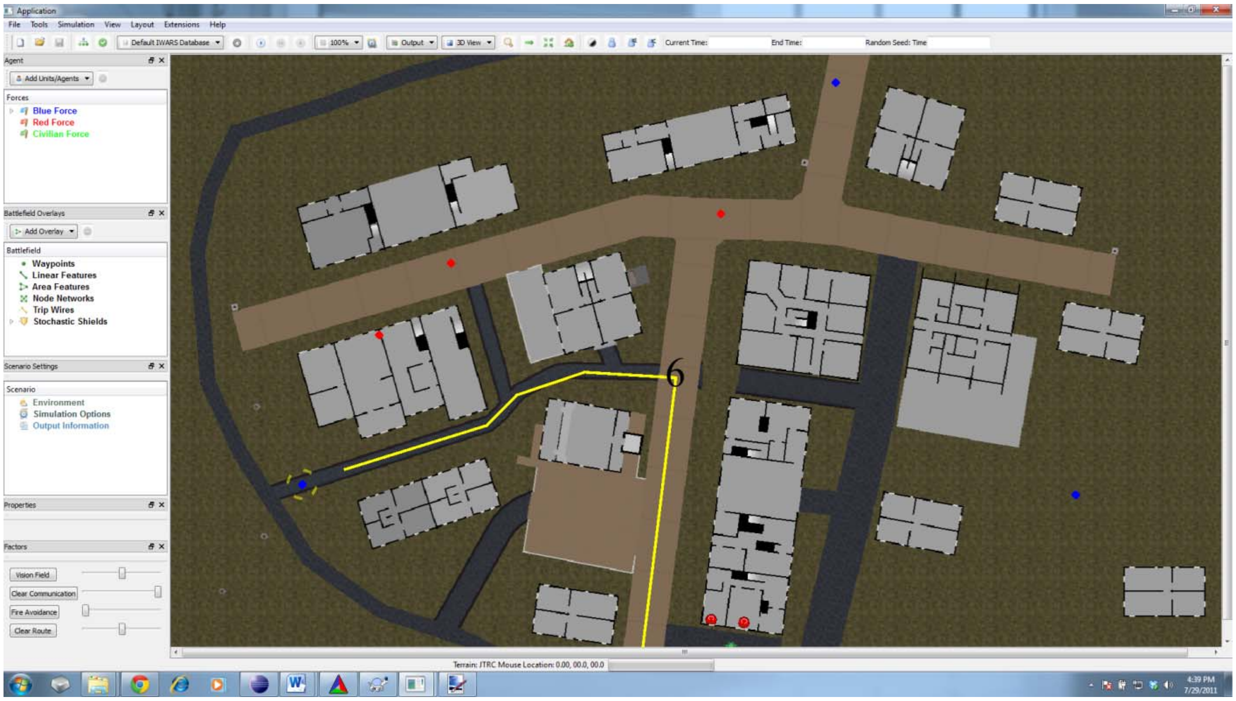


Figure 2: Routing Algorithm - IWARS

Additional information regarding agent overwatch and routing prototype can be found in the README_TacAid document within the software package.

4 Conclusions

4.1 Deviations from Planned to Actual Work

As research initiatives, SBIRs need tremendous flexibility in implementation so as to be responsive to the mix of customer needs and contractor capabilities. This is particularly true when a Phase I SBIR teams up with a contractor and government organization that have not worked closely together before. For this SBIR, Dignitas' Phase I proposal described a wide range of planned tasks that were not addressed based upon a better understanding of the customer's objectives for Phase I. However, the general vision outlined in the Phase I proposal could provide value, based upon NSRDEC's long-term interests. This section reviews objectives from the Phase I proposal that were not focused upon, and describes how those actions could still prove beneficial in Phase II and beyond.

Proposal Goal: Demonstrate common algorithms in use across multiple SAF systems and multiple platforms, including mobile and embedded (e.g. Command, Control, Communications, Computers, and Intelligence (C4I)) devices.

Phase I efforts focused on IWARS and related algorithm development. The concept of using a common algorithm for constructive, virtual, and operational use cases still seems valid as a Phase II objective, should NSRDEC be interested in pursuing that objective. This objective was intended to leverage Dignitas' experience in working across a wide range of systems, both for reuse (i.e. bringing good ideas into IWARS) and for commonality (i.e. providing the same algorithm / implementation to benefit multiple applications).

Since the proposal was written, Dignitas has had even more success in this area. For example, the Tactical Terrain Analysis (TTA) application, originally developed to provide support for mission planning and mission rehearsal, has drawn the interest of the live training community. At present, Dignitas is applying their mobile device experience to develop an app that will allow Observer/Controllers at live training ranges to control targets from an Android Tablet (Motorola Xoom) rather than from a range tower only. The TTA app has also been transitioned to the United State Military Academy.

Proposal Objective: Demonstration of multiple algorithms

Indirectly, this goal was met through algorithm description of two services rather than prototyping. Part of Dignitas' vision here was to leverage algorithms used in other Army applications, such as OneSAF, and apply them for the benefit of IWARS. However, this may require some experimentation to understand practical limitations of algorithm reuse, everything from programming language considerations to architectural questions like data access. A Phase II effort looking at this could provide the IWARS community a boost through algorithm reuse from other applications with minimal investment. If IWARS source code cannot be made available to Dignitas, then partnerships could be formed with other contractors to help them understand the ideal algorithms for reuse from other Army SAFs, including documenting the algorithm for reuse in the methodology documentation.

Proposal Objective: Demonstration across domains

The objective here was to demonstrate the same algorithms operating in constructive and virtual contexts, given Dignitas' strong connection to both domains (albeit outside of the IWARS / NSRDEC areas of effort). The algorithms described in the Phase I methodologies could be used in both a constructive and virtual context.

Proposal Objective: Wireless dynamic environment

This effort would have demonstrated the ability to modify geospatial data on a mobile device used for terrain analysis. This effort was the most distant from NSRDEC's objectives for this project, and is thus not discussed further.

As seen above, Dignitas' original proposal was heavily focused on prototyping and demonstration with a particular emphasis on a broad-based impact across domains and various Army applications. As NSRDEC's objectives for this SBIR were better understood, effort was redirected to understanding the basics of how IWARS works and how methodologies are described. Then the question was approached of how advanced terrain reasoning algorithms could be applied to enhance IWARS as described in Section 3.1.

4.2 Verification & Validation (V&V)

Team Dignitas has years of experience in the V&V process while working various programs of record such as CCTT, Synthetic Environment Core (SE Core) and OneSAF. The team understands clearly the importance of this process as it pertains to algorithms and methodologies within CGF systems such as IWARS. From reference (http://en.wikipedia.org/wiki/V&V), V&V is the process of checking that a product, service, or system meets specifications and that it fulfills its intended purpose. Verification is mainly a *quality control* process to evaluate if a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. This is often an internal introspection of the process used in the development of the capability. Validation is *quality assurance* process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

4.2.1 Capabilities

Specifically for this effort, three aspects of implemented capabilities undergo V&V. First the assumptions and approach of any algorithm methodologies must go through V&V. Second, the underlying data and implementation of the algorithms that are to be implemented need to undergo V&V. Third, the information that is to be used as input to the algorithms when emulating real-world tactical aids needs to undergo V&V.

The first aspect of V&V ensures that any development of algorithms, approaches, or math models is a valid representation of the real world. With any algorithmic representation of real world phenomena, there are inherent assumptions or approaches that generate some level of difference (or error). There are two levels of examination related to the method's resolution and its fidelity. Resolution pertains to how much of the real world phenomena is actually represented by the algorithm or approach described in the methodology. Its consideration should focus on what is meant to be achieved by the approach and if the resulting effects accomplish the original purpose. Fidelity pertains to how realistically (or accurately) the real world phenomena is represented by the algorithms and approaches described in the methodology. Its consideration should focus on the potential of injected error and its effects on accuracy relative to need.

The second aspect of V&V is the traditional case of V&V of constructive algorithms. In this aspect, any data sources, data, algorithms, and implementations of a capability need to be examined. For each capability, a validation authority or subject matter expert (SME) needs to be identified based on their recognized credentials. Past work with systems that underwent V&V confirmed that the Army Materiel Systems Analysis Activity (AMSAA) is both a recognized source of valid data and algorithms as well as an implementation validation agency. AMSAA would be a highly credible agency to support V&V activities for any algorithm implementations.

The third aspect of V&V is, at some level, a departure from a nominal constructive analysis algorithm. However, it parallels many of the efforts Dignitas has worked related to the development of validated behaviors. In these cases, prior systems have relied on SME input on the behavioral definitions of tasks as well as the use of Command and Control (C2) and Intelligence, Surveillance and Reconnaissance (ISR) capabilities. For this aspect, the NSRDEC may well be the appropriate agency to specify and verify the expected available input data and intended use. At the very least, NSRDEC could be the central broker of several SMEs to provide full coverage. One of the first aspects of any Phase II effort to consider V&V would be the determination of valid verification agencies.

4.2.2 Notional Process

The following sections describe about what inherent activities must occur in the process to execute a V&V process.

4.2.2.1 Verification

Verification is mainly an effort to make sure the capability was developed in a proper manner. This includes a few steps in the development process. First the effort must ensure that any specified and agreed process was followed. Second it must ensure that correct data is being used for the system. Third, it must ensure that any algorithms or sensitized data were developed properly. Lastly, it must ensure that data is being used properly and no errors are introduced.

There are a few ways this verification can be performed. First, there needs to be an examination of process execution artifacts for peer reviews and development standards (e.g., coding standards, architecture standards, security standards). The validation authority should also conduct code review for standards and requirements coverage. In addition, the authority should conduct a code review for accuracy of algorithm representation in code and data.

4.2.2.2 Validation

Validation is mainly a combination of implementation inspection coupled with collection of quantitative data related to the functional performance of a capability under a given test condition. The validation effort must ensure that any algorithms were rendered correctly and that data is correctly used. The effort must ensure that the implementation provides accurate compliance with performance specifications. This typically entails comparison of expected results with given inputs and may rely on gold standard data and/or scenario.

There are a few ways that validation can be performed. First, the validation authority should perform code inspections to determine that the algorithm is properly implemented. This usually requires traceability in the implementation back to design decisions, requirements, and any algorithm methodologies. The authority should also inspect the approaches used to manage and use data to ensure no errors can be introduced. Typically, this can entail some level of execution and data collection of performance to "gold standard" data set and scenario. In addition, either the contractor or authority will generate tests targeting specific performance specifications or data collection plans. Each of these tests highlight specific requirement(s). If the contractor is part of the test development, there needs to be concurrence with customer representative during test development (i.e., AMSAA representative). In addition the contractor may assist in ensuring the test results accuracy concurrence with the customer representative.

4.3 Next Steps

This Phase I effort provided Dignitas the opportunity to learn more about IWARS functionality, understand the algorithm methodology process, and present two methodologies for advanced terrain analysis services to help automate IWARS scenario generation and runtime functionality. This provides a foundation upon which Dignitas can provide far greater value to NSRDEC.

The most logical path forward is to continue with the path laid out during Phase I, that is development of methodologies and implementation into software as appropriate. Below are just a few examples of areas where Phase II could go; a Phase II proposal would include more detail as well as additional conceptual threads for NSRDEC to pick from.

4.3.1 Methodology Development

Dignitas can provide to NSRDEC the opportunity to leverage algorithms, concepts, and even software from large Army Modeling & Simulation investments in CGF systems, such as OneSAF. Continuing the Phase I effort, Dignitas can carry forward with definition of a wide range of methodologies, both in terrain services and other behaviors. For this work, Dignitas can leverage its extensive experience with Simulation Networking (SIMNET) SAF, ModSAF, Joint Semi-Automated Forces (JSAF), CCTT SAF, OneSAF, and more. Beyond Dignitas direct experience with algorithms from these systems, Dignitas has a wide range of experience with Combat Instruction Sets (CIS) and Physical Knowledge Acquisition documents (PKADs) based upon their critical roles on the CCTT and OneSAF programs.

A much higher rate of methodology generation is anticipated in Phase II. In Phase I, significant effort was invested in learning IWARS, understanding customer preferences, and learning the style of how methodologies are developed. In addition, if existing SAF experience is leveraged, methodologies can be created more quickly. The two primary methodologies developed in Phase I were new algorithms developed from scratch.

4.3.2 General IWARS Functional Enhancements

Dignitas' experience with a wide range of Modeling and Simulation systems, especially CGF systems, would provide a springboard for a broad range of enhancements to the IWARS system to enhance usability, scenario generation support, and automation of entity movements. Other improvements could include the ability to modify the simulation environment (e.g. show damage to buildings, for example). Dignitas is presently working on functionality that allows IED mounds with wires or ant trails to be placed in visual scenes in a blended, realistic manner. This type of functionality could be integrated into the Delta3D engine to enhance the IWARS environment.

Dignitas is open to working with NSRDEC and other contractors in whatever configuration gets the job done. For example, Dignitas could focus on methodology development, work directly with IWARS development through APIs, or implement updates in IWARS using a working copy for experimentation. All of these approaches would protect the IWARS V&V'ed baseline from arbitrary updates.

4.4 Dignitas Benefits for Phase II

This Phase I effort provided an opportunity to better understand NSRDEC and IWARS. Similarly, it was an opportunity to describe Dignitas' unique company capabilities to NSRDEC. For example, the Phase I proposal provided past performance information. This section provides some supplemental high-level descriptions of how Dignitas could benefit a Phase II SBIR effort.

A key element of all of Dignitas' SBIR efforts is a focus on Government Purpose Rights development. This means that all development efforts undertaken are specifically intended to be extended for use in Department of Defense (DoD) applications and to be made fully available to any government program (and government contractor) who needs it. This openness means that the results of Dignitas' SBIR efforts can easily be provided to other vendors (e.g. those who provide overall maintenance of a product baseline) without rights encumbrances. This approach is beneficial to customers, representing a sharp contrast to companies who try to keep software close hold so as to avoid competition and try to maintain a monopoly on software.

As a natural extension of this philosophy, Dignitas is often asked to play an "honest broker" or 3rd party integration role for government customers or contractors. For example, Dignitas supports or leads the integration of OneSAF into multiple major Army applications running across multiple domains, including Homestation Instrumentation Training System (HITS) in the live domain, Conduct of Fire Trainer – Situational Awareness (COFT-SA) in the virtual/embedded domain, and functional enhancements for U.S. Army Training and Doctrine Command (TRADOC) Intelligence Support Activity (TRISA) in the constructive domain. As another example, Dignitas is widely recognized by the Program Executive Office for Simulation, Training, and Instrumentation (PEO-STRI) and Army Research Lab - Simulation and Training Technology Center (ARL-STTC) communities as being successful with transition of research concepts into major production programs. Dignitas is actively working on integration of research capabilities into Army trainers large (CCTT) and small (Construction Equipment Virtual Trainer (CEVT)).

Dignitas has an extremely high success rate across all areas of SBIR endeavor. Of the 5 Phase I's Dignitas has completed to date, 4 have gone to Phase II. Of the 4 Phase II's, 3 have continued on to Phase III funding before the first year of the Phase II effort was even completed. In addition, Dignitas has received Fast Track and Enhancement funding. Three of 4 Phase II efforts have received more matching funds from other sources than the total SBIR office investment in Phase I and Phase II. All of Dignitas' SBIRs demonstrate a high success rate in implementation with real, usable, and delivered capabilities. Where possible and appropriate, Dignitas conducts extensive outreach efforts during SBIR execution to maximize reuse across potential DoD users. Dignitas' SBIR technology has been applied to use cases as diverse as major training simulations (CCTT Reconfigurable Vehicle Simulator (RVS)), integration into C4I devices (Real-time Adversarial Intelligence and Decision-making (RAID)/ Force XXI Battle Command Brigade and Below (FBCB2)), use in U.S. Military Academy courses (West Point orienteering), and many exchanges across research efforts.

This document reports research undertaken at the
U.S. Army Natick Soldier Research, Development and
Engineering Center, Natick, MA, and has been
assigned No. NATICK/TR- 12/019 in a
series of reports approved for publication.

List of Acronyms

Acronym Name	Definition
AMSAA	Army Materiel Systems Analysis Activity
AOP	Agent Overwatch Position
API	Application Programming Interface
ARL-STTC	Army Research Lab - Simulation and Training Technology Center
ARP	Agent Route Planner
C2	Command and Control
C4I	Command, Control, Communications, Computers, and Intelligence
CCTT	Close Combat Tactical Trainer
CETS	Common Embedded Training System
CEVT	Construction Equipment Virtual Trainer
CGF	Computer Generated Forces
CIS	Combat Instruction Sets
COFT-SA	Conduct of Fire Trainer – Situational Awareness
C&C	Cover and Concealment
DoD	Department of Defense
FBCB2	Force XXI Battle Command Brigade and Below
FOV	Field Of View
GUI	Graphical User Interface
HITL	Human In The Loop
HITS	Homestation Instrumentation Training System
IED	Improvised Explosive Device
ISR	Intelligence, Surveillance and Reconnaissance
IWARS	Infantry Warrior Simulation
JSAF	Joint Semi-Automated Forces
LSE	Location Selection Engine
LOS	Line Of Sight
ModSAF	Modular Semi-Automated Forces
NSRDEC	U.S. Army Natick Soldier RD&E Center
OneSAF	One Semi-Automated Forces
OTB	OneSAF Testbed Baseline
PEO-STRI	Program Executive Office for Simulation, Training, and Instrumentation
PKAD	Physical Knowledge Acquisition Document
RAID	Real-time Adversarial Intelligence and Decision-making
RVS	Reconfigurable Vehicle Simulator
SAF	Semi-Automated Forces
SBIR	Small Business Innovation Research

SE Core	Synthetic Environment Core
SIMNET	Simulation Networking
SME	Subject Matter Expert
TRADOC	U.S. Army Training and Doctrine Command
TRISA	TRADOC Intelligence Support Activity
TTA	Tactical Terrain Analysis
V&V	Verification and Validation

Bibliography

Campbell, C., Hull, R., Root, E., and Jackson, L. “Route Planning in CCTT” In Proceedings of the 5th Conference on Computer Generated Forces and Behavioral Representation. University of Central Florida. (1995)

Darken, Christian J., “Visibility and Concealment Algorithms for 3D Simulations”, MOVES Institute and Computer Science Department, Naval Postgraduate School, Monterey, CA

Li, Qun, DeRosa, Michael, Rus, Daniela “Distributed Algorithms for Guiding Navigation across a Sensor Network,” Department of Computer Science, Dartmouth College, Hanover, New Hampshire.

Reece, Douglas A., Kraus, Matt, Dumanoir, Paul, “Tactical Movement Planning for Individual Combatants,”

Vinson, N.G., “Design Guidelines for Landmarks to Support Navigation in Virtual Environments, In Proceedings of CHI ’99, Pittsburgh, PA. (May 1999)

Appendix A Route Planner Methodology

A.1. Route Planner Methodology Overview

The simple process of going from point A to point B can be complicated when external factors are introduced. Factors such as terrain conditions, impassable obstructions (e.g. large lakes), and hostile locations can shift the most practical route (i.e. a straight line) into an intricate set of directions.

Simulation systems tend to route in two different situations: on networks (e.g. roads, rivers, hallways) and cross-country (e.g. open terrain). Although conceptually different, many of the same factors used to determine the optimal route apply to both and are essential to the calculation.

Many important pre-computation parameters are needed for a methodology to determine the optimal route (or a set of optimal routes provided to a user for final determination) in an automated fashion. Among these parameters are the start and stop points of the route, along with waypoints (intermediate route points that must be traversed). Each conditional parameter that is inputted will be used to generate possible routes.

The Agent Route Planner (ARP) methodology calculates the optimal route by evaluating a suite of routing factors (e.g. enemy locations, areas of cover and/or concealment) to determine its results. It presents to the user a calculated optimal route, and other alternative routes, along with the rationale (i.e. factor scores) on why these spots were selected. The methodology is configurable and expandable to allow a user to define their criteria for route selection.

A.2 Assumptions

The methodology assumes the underlying terrain model is capable of the following services: height of terrain, feature lookup, collision detection, and ray tracing.

A.3 Use Cases

The ARP methodology is envisioned to be used in two different scenarios: as a part of a constructive simulator, and as a service present on an operational device (e.g. smartphone capability).

A.3.1 Constructive Use Case

The ARP methodology will prove to be a valuable service in constructive simulation. The following three use cases present different envisioned scenarios for the ARP methodology.

A.3.1.1 Constructive Use Case 1 - Mission Planning

In the pre-exercise mission planning use case, the methodology will allow a user to determine the optimal route, scored against their tailored factors, to optimally route their agent from route start to completion, traversing through all selected waypoints.

- User defines inputs needed for factors.
- User selects factors used to determine agent ARP.
- The user configures the weighted factors and their properties.
- The methodology calculates optimal routes.
- User investigates routes, and their scores, to determine final route (if any).

A.3.1.2 Constructive Use Case 2 – Calculating ARP during Simulation Execution

This use case is similar to the previous use case, except that ARP is calculated during scenario execution. In order to meet real-time, or at least the user's execution time requirements, factors may have to be tailored due to performance.

- User defines inputs needed for factors.
- User selects factors used to determine agent ARP.
- The user configures the weighted factors and their properties.
- The methodology calculates optimal routes.
- User investigates routes, and their scores, to determine final route (if any).

A.3.1.3 Constructive Use Case 3 – Determining ARP for A User Inputted Route

This factor executes in a similar manner to the previous use cases. The methodology scores a user created route. The user selects the route of interest, and all of the factors are executed on the selected route, and a score is calculated. Each factor's score is presented back to the user, allowing the user to determine if their route is a desired route.

- User defines route.
- User defines inputs needed for factors.
- User selects factors used to determine agent ARP.
- The user configures the weighted factors and their properties.
- The methodology calculates score for selected route.

A.3.2 Operational Use Case

The envisioned use of this methodology is not only for constructive simulation. A service to determine the optimal route between a set of points for an agent is also extremely valuable for an operational service "in the field."

A.3.2.1 Operational Use Case 1 – Mission Planning

In the mission planning use case, the methodology will allow an agent to determine an optimal route which meets their tailored factors.

- User defines inputs needed for factors.
- User selects factors used to determine agent ARP.
- The user configures the weighted factors and their properties.
- The methodology calculates optimal routes.
- User investigates routes, and their scores, to determine final route (if any).

A.3.2.2 Operational Use Case 2 – In Theater ARP

This use case is similar to the mission planning use case except for in this scenario the agent is already in theater and is using the methodology to determine an optimal route from their current location.

- User selects factors used to determine agent ARP.
- User adjusts weighted factors and their properties based on actionable real time intelligence.
- The methodology calculates optimal routes for transition.
- User investigates routes, and their scores, to determine ARP.

A.4 Methodology Inputs

Route Points:

- Route entry point
- Route release point
- Route waypoints

Of the agent:

- Sensor (type, range, attenuation factor due to distance, field of view).
- Size (Height).
- Posture (available postures)
- Speed
- Equipment/load
- Start location.

Environmental:

- Terrain elevation values.
- Volumetric terrain features.
- Ambient lighting.

Other:

- Location of teammates.
- Perceived enemy locations.

A.5 Methodology

The ARP methodology is a composable service. The service contains and manages a suite of “factors” which can be utilized to determine the optimal route between a series of points. Optimal, in this case, does not necessarily mean shortest or fastest, but is defined by a series of factors that are applied to each route segment. The methodology works on both networked and cross-country routes. It works in three phases: network composition phase, network traversal engine, and factor suite. The basic flow of the service is as follows:

- Service configured by caller.
- Service executed.
- Establishment of traversal network.
- Traversal engine executes.
- Network arcs are scored.
- Optimal route (with factor scores and metadata) is returned to the caller.

A.5.1 Network Composer

The routing service works to calculate the optimal path in a network. This means, for the service to work, a network has to be established. For network routing (roads, rivers, urban corridors, and building interiors) this is easy, the system will rely on the terrain network available in the terrain model. For cross-country routing, a network needs to be established. With all aspects of this methodology, the network composer is composable, allowing for any network composition algorithm to be used. The following describes the initial network composer implementation.

Cross-country regions will be rasterized to create a regular grid. Each cell of the grid will represent a node in the graph. Each node will have an arc from itself to all of its eight adjacent nodes (except for boundary cases). The arcs are checked to make sure they are traversable. If they are not traversable (blocked by impassable terrain feature, steep slope, etc.) the arc is eliminated.

A.5.2 Traversal Engine

Once the arc-node graph is established, it is traversed via the traversal engine. As with all other aspects of this methodology, the traversal engine is composable, allowing for any network traversal engine to be used. The A* service will be implemented as the initial traversal engine.

A.5.2.1 A* Engine

A* uses a best-first search to find the optimal route from start to stop nodes (using waypoint nodes if supplied). It uses a factor based cost heuristic function to calculate traversal order. The cost heuristic is a sum of two functions: the path-cost function, which is the cost from the starting node to the current node, and an admissible "heuristic estimate" of the distance to the goal. Note: the heuristic estimate must not overestimate the distance to the goal. Thus, for an application like routing, it will be represented by the straight-line distance to the goal, since that is physically the smallest possible distance between any two points or nodes.

The cost of each node will be calculated by the available factor suite.

A.5.3 Factors

Factors are used to score the arcs of the traversal graph. Each factor calculates a score between 0 and 1, with 0 being the worst, and 1 being the best. Each factor is configured with the same set of data: agent in question, active arc, perceived enemy, and terrain/environment model. Although each factor will receive the same data, it will only use the data that is applicable to its needs.

For ARP, factors can be developed into an arc score based on many different criteria. Some of the most common factors for routing are distance, speed, agent posture, cover/concealment, and terrain composition. As with all other aspects of the methodology, the system is configurable to account for many different types of data to help determine the optimal route.

A.5.4 Tailoring ARP

The user will have the capability to configure their instance of the ARP service immediately prior to execution. They will be able to define, add, and select which factors will execute for their query. Beyond that, the user will be able to add a weighting to each factor to add priority to the factors that are the most important. The user will also be able to define a minimum score to a factor, which will completely eliminate a query if the calculation falls below it.

A.5.5 Default Factors

Default factors will be developed to calculate the ARP. As previously stated, the ARP methodology is a composable service allowing for the suite available factors to grow in the future.

A.5.5.1 Cover and Concealment of Route

A major factor in determining a desirable route is the safety of the agent as they traverse the route. Cover and concealment services are generally provided by the underlying system's terrain model to calculate if an agent is covered (i.e. not able to be shot) and/or concealed (i.e. not able to been seen) from a certain location. This methodology will use the system's cover and concealment capability (if available) as a factor for ARP. The cover and concealment service will be called for each route arc. The service will calculate the arc's cover/concealment score. Cover and concealment may need to be called multiple times (once for each posture) to determine how cover may impact traversal rate of the graph.

A.5.5.2 Distance of Route

Total distance of route is an important factor for routing. A longer route may adversely affect a potential route in many ways, including total travel time, overall energy required to traverse the route, and continued exposure to hostile situations. In short, the longer the route is, the more opportunities there are for bad things to happen.

This factor will calculate the overall distance of a route. Each route is composed of a series of graph arcs. The final distance of the route is the summation of all of the individual arc distances.

A.5.5.3 Traversal Time

Traversal time, in many ways, goes hand in hand with distance of the route, the longer the route, the longer it takes to traverse. However, there are other factors which do not necessarily make this a one to one comparison. Some terrain features (e.g. swamps, mud) may slow down the rate at which an agent can traverse an area. Changes in terrain elevation also impact traversal time, uphill climbs will probably slow down traversal, where downhill traversal may actually speed up traversal time.

Each arc will be calculated for traversal speed based on the agent in question. Distance of the arc, terrain features, terrain slope, and agent composition (along with posture) will all be used to determine the overall traversal rate.

A.5.5.4 Terrain Obstacles

In the previous section it was discussed how some terrain features and/or terrain elevation changes could adversely (or favorably) affect traversal rate and overall time. There are some terrain obstacles, however, that do more than merely slow down an agent. A wide river for example, may simply be impassable, and completely eliminate a potential route.

Each route arc will be checked for features or terrain characteristics which adversely affect a route. The more difficult a terrain obstacle is to traverse the higher a score it will be given, with an impassable obstacle given the maximum score.

A.5.5.5 Danger Areas

This factor will calculate possible danger areas for a selected route. It will calculate this result by conducting the Agent Overwatch Position (AOP) methodology for the enemy units using the selected route for the methodology’s input. The AOP calculates the optimal position for an entity insertion point within an area of interest based on weighted factors and their properties (field of vision, crossfire/friendly fire avoidance, and clear route). For this factor instead of calculating the optimal point for an entity insertion, the AOP algorithm will be used to determine possible threat locations within the bounding area of a possible route. If threat positions are located within a specified distance to a projected route then a negative score against an arc can be imposed.

A.6 Methodology Outputs

It is essential for services to supply to the operator more than just the “correct” answer. Metadata and alternative solutions should be presented to the operator to allow them to understand the rationale for the solution. Because of this, the output of the ARP is not just a single route.

The methodology results in a sorted list (sorted from best to lowest score) of route arcs and metadata. Each arc will have its world coordinates and its overall factor score. Additionally it will have a list of factor records, containing the factor, its score, and any associated metadata for that factor.

It will be up to the calling application (for instance IWARS) to take the results of this methodology and display the results in an intuitive, meaningful, and graphical way.

This page intentionally left blank

Appendix B Agent Overwatch Position

B.1 Agent Overwatch Position Overview

The agent overwatch position methodology determines the optimal position for a combatant taking many factors into consideration including the assigned area of interest (AI) and cover from perceived hostile locations. The methodology scores potential locations in an area of interest and results in a suggested optimal location, along with alternative locations, with the rationale on why these spots were selected. The methodology is configurable and expandable to allow a user to define their criteria for location selection.

B.2 Assumptions

The methodology assumes the underlying terrain model is capable of the following services: height of terrain, feature lookup, collision detection, and ray tracing.

B.3 Use Cases

The agent overwatch position methodology is envisioned to be used in two different scenarios: as a part of a constructive simulation, and as a part of an operational situation awareness capability.

B.3.1 Constructive Use Case

The AOP methodology will prove to be a valuable service in constructive simulation. The following are three envisioned use cases for constructive simulation:

B.3.1.1 Constructive Use Case 1 - Mission Planning

In the pre-exercise mission planning use case, the methodology will allow a user to determine the optimal location, meeting their desired factors, to place their entity within a predefined area of interest.

- User defines area of interest.
- User selects factors used to determine agent AOP.
- The user configures the weighted factors and their properties.
- The methodology calculates optimal locations for overwatch.
- User investigates locations, and their scores, to determine AOP.

B.3.1.2 Constructive Use Case 2 – Calculating AOP during Execution

This use case is similar to the previous use case, except that AOP is calculated during scenario execution. In order to meet real-time, or at least the user's execution time requirements, factors may have to be tailored due to performance.

- User defines area of interest.
- User selects factors used to determine agent AOP.
- The user configures the weighted factors and their properties.

- The methodology calculates optimal locations for overwatch.
- User investigates locations, and their scores, to determine AOP.

B.3.1.3 Constructive Use Case 3 – Calculating AOP for Selected Location

Instead of determining the best location for an agent out of an area of interest, this use case uses the methodology to score a position of interest. The user selects the location of interest, and all of the factors are executed on the selected location, and a score is calculated. Each factor's score is presented back to the user, allowing the user to determine if their location is a desired overwatch position.

- User defines agent location of interest.
- User defines overwatch area of interest.
- User selects factors used to determine agent AOP.
- The user configures the weighted factors and their properties.
- The methodology calculates score for selected location.

B.3.2 Operational Use Case

The envisioned use of this methodology is not only for constructive simulation. A service to determine the optimal location for an agent is also extremely valuable for an operational service "in the field." For example, an agent may use this methodology to determine optimal observation locations for a threat (e.g. suspected sniper location) or to determine the likely locations of threat agents.

B.3.2.1 Operational Use Case 1 – Mission Planning

In the mission planning use case, the methodology will allow an agent, in real time or near real time, to determine the optimal location for overwatch.

- User defines area of operation.
- User selects factors used to determine agent AOP.
- The user configures the weighted factors and their properties.
- The methodology calculates optimal locations for overwatch.
- User investigates locations, and their scores, to determine AOP.

B.3.2.2 Operational Use Case 2 – In Theater AOP

This use case is similar to the mission planning use case except for in this use case an agent is already in theater. An agent will be able to input actionable intelligence (e.g. threat location) and use that to determine an optimal overwatch location.

- User defines area of operation
- User inputs intelligence information.
- User selects factors used to determine agent AOP.
- User adjusts weighted factors and their properties.
- The methodology calculates optimal locations for overwatch.

- User investigates locations, and their scores, to determine AOP.

B.3.2.3 Operational Use Case 3 – Reverse AOP to Calculate Likely Threats

This use case demonstrates using the AOP methodology to determine likely threat positions. This works in a similar fashion to the previous use cases, but in reverse. In this use case the user will select an area of operation, input weighted factors and their properties to determine possible threat locations (e.g. sniper locations).

- User defines area of operation.
- User selects factors used to determine threat positions.
- User adjusts weighted factors and their properties based on actionable real time intelligence.
- The methodology calculates threat locations.
- User investigates locations, and their scores.
- User determines on how to approach or avoid possible threat locations.

B.4 Methodology Inputs

Areas of interest:

- Rectangular area representing the area of interest.
- Rectangular area of possible overwatch positions.
- Rectangular area(s) of potential threat positions.

Of the agent:

- Sensor (type, range, attenuation factor due to distance, field of view).
- Size (Height).
- Start location.

Environmental:

- Terrain elevation values.
- Volumetric terrain features.
- Ambient lighting.

Team:

- Location of teammates.

Algorithm Configuration:

- Raster size.

B.5 Methodology

The Agent Overwatch Position (AOP) methodology is a composable service. The service contains and manages a suite of "factors" which can be utilized to determine the overwatch value at a particular location. The service contains a "Location Selection Engine (LSE)" which decomposes an area into a set of discrete locations. The basic flow of the service is as follows:

- Service configured by caller.
- Service executed.
- LSE initialized with proper parameters and executes.
- LSE executes factor suite and derives a score for location.
- LSE repeats process until the engine satisfies its exit criteria.
- List of overwatch positions (with factor scores and metadata) is returned to the calling application. The application will display the results in a reasonable fashion, most likely selectable icons on a map attached to its scoring matrix.

B.5.1 Location Selection Engine

As stated previously, the Location Selection Engine (LSE) is a composable element. This means the system can be defined by multiple LSEs and chosen at execution time which to operate with. In general, a defined LSE instance will operate on the available service data and can be as simple as a single method, or call to an external service engine itself.

An example of using an external service for the overwatch LSE would be locating only possible areas that have above a threshold cover/concealment score. The AOP methodology would pass in the area of interest and additional required data to the cover/concealment (C&C) service. The C&C service would return a list of locations. The LSE would iterate over each returned location and calculate its AOP score.

This architecture allows users to tailor the service to meet their specific needs. Each complexity added to the AOP service will impact the execution performance. In some cases a user may not need as high of a resolution service and evaluate the tradeoffs which occur when tailoring the service. This architecture also allows for further research in different types of algorithms (such as genetic algorithms) to generate locations for the AOP methodology.

This architecture also is optimized to work in a multi-threaded environment. Each factor can execute independently of the results of another factor. Each factor can execute in parallel with another factor. This can allow for a tremendous performance improvement on systems which can support concurrent operations.

B.5.2 Factors

Factors are used to score a selected location for AOP. Each factor calculates a score between 0 and 1, with 0 being worst, and 1 being the best. Each factor is configured with the same set of data: agent in question, query location, overwatch area, potential threat areas, and handles to the environment (including terrain).

For AOP, factors can be developed to determine the best position based on many different criteria. The most obvious factor for overwatch is line-of-sight into an area. But the service is not limited to just that, other factors could look at criteria such as cover and concealment, distance to target, distance and route from current location to overwatch location, etc. Like the LSE, the system is configurable (both by selected factors, and weighing of each factor's results) to take into account the many different types of data to help determine the ideal location.

B.5.3 Tailoring AOP

The user will have the capability to configure their instance of the AOP service immediately prior to execution. They will be able to define, add, and select which factors will execute for their query. Beyond that, the user will be able to designate a weight to each factor to prioritize to the factors that are the most important. The user will also be able to define a minimum score to a factor, which will completely eliminate a query if the calculation falls below it. From a user interface standpoint, factors may be presented with checkboxes on the side, allowing an operator to add or remove factors as needed. The dialog could also provide a means for operators to assign a ranking between the factors based upon their objectives.

B.5.4 Default Factors

Default factors will be developed to calculate the AOP.

B.5.4.1 Observable Area

Probably the key factor of AOP, the Observable Area factor determines the overall visibility from the selected position to the area of interest. This is accomplished by transforming the agent's potential maximum distance where a line-of-sight could exist into a raster of pixels. A line-of-sight (LOS) ray is sent from the agent's view point to each pixel and its visible value is recorded.

The visible value is a percentage (value between 0 and 1) which represents the distance on the ray where the LOS was blocked. A value of 1 means that the LOS ray was not blocked. A value of 0 means that the ray was immediately blocked.

In order to achieve optimal results, some care has to be taken in construction of the field of view (FOV) raster. The initial raster is constructed as a spherical plane with each cell the agent's sensor range from the query origin. The raster is then intersected with the clipping planes of the region of interest; this is done to eliminate any checks for clear LOS beyond the area of interest. The same clipping cannot be done in the areas before the area of interest boundary, because an LOS obstruction prior to the area of interest boundary will manifest itself inside the area of interest. In fact, the earlier that the blockage occurs the more of an impact it will have on the entire scene.

Next, the top (max elevation) of the raster is modified to only check for a defined elevation off of the terrain. This is done to alleviate some of the false positive results that can occur by checking LOS too far above the terrain skin. Above a certain elevation, no (ground) combatant can exist, so it is pointless to check against it.

Finally, the raster is modified by clipping the bottom of the raster with the “base” terrain skin. The base terrain skin represents a flat plane of the lowest elevation in an area of interest. This is done so that terrain blockages of the lowest elevation do not adversely affect the score of the factor. An LOS ray that intersects the lowest level terrain signifies that the agent’s overwatch position can completely see the area of interest for that pixel, so the result should be 1.

An advance feature for the vision field service is to allow the user to define critical areas for visibility. For example, an agent’s primary focus may be on the defense of a river bank. One possible agent location may return a higher score because of the overall view of the overwatch area is clear. But in this situation, the agent’s most critical point of interest is behind the gatehouse of the bridge. The overwatch location, although good for the general case, has very poor visibility of the gatehouse. If the user specifies that the gatehouse is a critical area, a different location, with a clear view of the gatehouse, will be scored higher, and thus, selected as the primary location.

After each pixel has been checked, a final score is tabulated by averaging the value of each pixel. The final score is then returned to the caller.

B.5.4.2 Team Positioning

The team positioning service scores the current agent’s location in regards to the agent’s proximity to other members of their squad. The factor will be capable of being configured to account for other entities, including teammates, squad leaders, platoon leaders, etc.

B.5.4.3 Crossfire/Friendly Fire Avoidance

This factor checks to see if any other friendly agent is located in the active agent’s field of view and weapon range, when facing toward the area of interest.

B.5.4.4 Clear Route

An important factor for an overwatch position is whether an agent travels from their current location to the overwatch position safely. The route safety factor will call the routing service between the agent’s current position and the active query position. The routing service will return a value representing the overall score (safeness) of the route.

B.5.5 Default LSE

Like other parts of the architecture, the LSE is composable and can be changed at any time to meet the requirements of the user. The default LSE will rasterize the potential location area and check points in a regular grid. The engine will run and check every cell before exit.

B.6 Methodology Outputs

It is essential for services to supply to the operator more than just the “correct” answer. Metadata and alternative solutions should be provided to the operator to provide them the rationale for the solution. Because of this, the output of the AOP is not just a single position.

The methodology results in a sorted list (sorted from best to lowest score) of location and metadata. Each location will have its world coordinate and its overall factor score. Additionally it will have a list of factor records, containing the factor, its score, and any associated metadata for that factor.

It will be up to the calling application (for instance IWARS) to take the results of this methodology (a sorted list of locations with metadata) and display the results in an intuitive and meaningful way.